LEARNING PLANNING HEURISTICS THROUGH OBSERVATION

Charlie Dolan
Michael G. Dyer

April 1986
CSD-860071

Learning Planning Heuristics through Observation*

Charlie Dolan
Artificial Intelligence Lab
Computer Science Department
UCLA
and
Hughes Aircraft AI Center

Michael Dyer
Artificial Intelligence Lab
Computer Science Department
UCLA

Topic Area: Learning
Key Words: planning, constraints, chunking

Address:
UCLA
AI Lab
Boelter Hall
Los Angeles, CA, 90024
ARPA: cpd@UCLA-LOCUS.ARPA

Abstract

This paper discusses a method for learning thematic level structures, i.e. abstract plan/goal combinations, by observing the bad planning behavior of narrative characters and combining old descriptions of planning errors to create new, abstract structures. The learning method discussed is a one-trial, schema acquisition method, which is similar to DeJong's [DeJong, 1983]. The method used involves taking schemas for planning situations that are found in an actual narrative situation, and using constraint-based causal reasoning to construct a new schema which better characterizes the situation. This work is part of the MORRIS project at UCLA [Dyer, 1983]. The planning situations are represented using Thematic Abstraction Units (TAUs) [Dyer, 1983].

* This work was supported in part by a grant from the Hughes Aircraft AI Center and the Keck Foundation.

1 Introduction

In the real world, tasks cannot always be accomplished by using simple subgoal partitioning and recursive problem analysis. Both real world agents and narrative characters often must apply plans that require cooperation from other agents, adjust plans that conflict with an agent's concurrent goals, and manage plans which contribute simultaneously to more than one goal. A classification of real world plans is found in [Schank and Abelson, 1977]. A taxonomy of goal/plan interactions can be found in [Wilensky, 1978]. In addition, successful real world planners often use adages to guide them in avoiding bad plans. Adages warn against both specific and general planning errors. *Poor Richard's Almanac* [Franklin, 1733-1758] gives many examples of adages, such as "A stitch in time saves nine" (error avoidance), and "Don't burn your bridges behind you." (error recovery).

These metaphorical admonitions are similar in function to Sussman's planning critics [Sussman, 1973]; however, they are much more varied than the fixed number of very general critics Sussman suggests, and as this paper shows, they are learnable. It is important to represent planning errors, not only to critique plans, but also so that in counter-planning situations [Carbonell, 1979], a planner can try to trick another agent into making a planning error.

Dyer [1983] showed how a class of planning errors could be represented by Thematic Abstraction Units (TAUs), and how these planning errors might be recognized in stories. This paper will present a representation for planning error recognition that also facilitates the combination of planning descriptions into new thematic structures. The combination method requires an example narrative situation that contains a new planning error. The example is conceptually analyzed to discover whether known planning errors can be combined into new structures and then to establish the links between the component structures that make up the newly discovered planning error.

In addition to the representations for TAUs, goals, and plans mentioned above, the examples here also rely on Schank's Conceptual Dependency theory [Schank, 1972]. Other relevant work on memory organization includes [Schank, 1982], [Kolodner, 1980], and [Lebowitz, 1980].

2 An Example Planning Situation

Here we will see a planning situation that contains three planning errors whose descriptions are already known to the system. From this situation we will see how we can generate two specializations of planning errors, and one novel planning error construct. The situation is taken from Aesop's fables; the version below is taken from Bewick's collection [Bewick, 1973].

Other work dealing with Aesop's fables is TALESPIN [Meehan, 197x], a program which generates stories by simulating a character's planning and problem-solving behavior. TALESPIN lacked a theory of planning error analysis. As a result, although **The Fox and the Crow** story was generated, it was accomplished only by using a pre-defined, non-extensible representation of that specific planning situation.

The Fox and the Crow

> The Crow was sitting in the tree with a piece of cheese in her mouth. The Fox walked up to the bottom of the tree and said to the Crow, "Crow what a beautiful voice you have; please sing for me." The Crow was very flattered and began to sing. When she did, the cheese dropped out of her mouth. The Fox grabbed the cheese and ran away laughing.

Note that this story can be looked at in two ways: 1) as an instance of bad planning on the part of the Crow and 2) as an instance of good planning on the part of the Fox.

The first of the three planning errors we will discuss is the simplest. When the Crow sings, she does not realize that she is already using her mouth to hold the cheese. This planning error is characterized at an abstract level by TAU-CONF-ENABL (confused enablement). The full representation for TAU-CONF-ENABL is given below.

In each TAU, the representation of a planning error situation consists of two parts: (1) the **binding-spec**: a list of conceptual patterns which occur in the story; (2) the **constraints**: a list of logical constraints among the patterns occurring in the **binding-spec** and other concepts from the story.

```
TAU name TAU-CONF-ENABL
  binding-spec
 [?standing-goal (P-GOAL actor ?x
                 obj (POSS-BY actor ?x
                              obj ?y)
                 manner FAIL)]
 [?interfering-goal2 (GOAL actor ?x)]
 [?mistake (ACT actor ?x)]
  constraints
 intention(?interfering-goal,?plan),
 realization(?mistake,?plan),
 resulting(?mistake,?disabling-state),
 achievement(?desirable-state,?standing-goal1),
 disablement(?disabling-state,?desirable-state).
```

The abstract situation this structure characterizes is one where an agent has a goal, **?standing-goal**, which has failed and where the goal was to preserve possession of some object. The cause of the goal failure is an act, **?mistake**, which attempted to accomplish another goal, **?interfering-goal**.

This TAU serves as a description of a planning error and as such can be used as planning advice. Previous research has shown how constraints can be used for problem solving [Steele, 1980]. The constraints presented here are easily implemented as a logic program [Kowalski, 1979].

The representation presented here enables us to create new TAUs from existing ones. The processes of recognizing and indexing TAUs are covered more fully in [Dyer 1983] and [Dolan 1984]. Dolan [1984] also covers the comprehension process model which allows the recognition of TAUs in this format when planning errors are encountered in situation descriptions.

As we mentioned above, **The Fox and the Crow** instantiates two other TAUs: (1) TAU-VANITY is the planning error of allowing personal vanity to dictate plan choice; (2) TAU-ULTERIOR is the planning error of not considering another agent's possible motives before acting. Both of these TAUs display an important characteristic of TAUs as planning heuristics; not only do TAUs provide admonitions against bad planning, but they can also be turned around and used as plans to try and force other agents into situations where they will make mistakes.

These TAUs can be combined to form new planning heuristics. There are two key problems in TAU acquisition:

1. How does a program know which TAUs to select and examine for combination attempts?

2. Once selected, how are TAUs actually combined to form new planning and indexing structures?

Both 1. and 2. are non-trivial. A sophisticated planner should have numerous stories indexed by multiple TAUs in memory. Attempting to combine TAUs arbitrarily would lead to combinatoric problems. Fortunately, memorable stories (such as Aesop's fables) are designed to give novel planning advice through illustrating planning errors. Thus, TAU selection can be governed by the following strategy:

WHENEVER two TAUs share concepts in an observed
planning situation,
TRY to combine them to form a novel planning construct

This heuristic can only be applied after reading a story. The comprehension of the story thus makes available TAUs for combination and indicates which concepts are shared. Even when two TAUs share concepts, however, they may not be able to be combined.

There are two ways to combine TAUs based on the way they share concepts:

(1) specialization,
(2) combination (chunking).

Recent work in specialization learning includes [DeJong, 1983] and [Kolodner, 1980]. Both workers formulate methods for creating new planning knowledge through specialization, but do not have a method extensible to chunking. Most research in learning by chunking has been in domains where there is no counterplanning [Laird, 1984].

3 Creating New TAUs through Specialization

The type of sharing which exists between TAU-ULTEROR and TAU-CONF-ENABL is call **containment**. In order to see how containment works, examine the representation for TAU-ULTERIOR given below,

```
TAU name TAU-ULTERIOR
    binding-spec
     [?standing-goal (GOAL actor ?x
                           status FAILED)]
     [?desirable-state (STATE)]
     [?mtrans (MTRANS actor ?y
                      to ?x
                      obj ?z)]
     [?mistake (ACT actor ?x)]
    constraints
     achievement(?desirable-state,?standing-goal),
     motivation(?z,?interfering-goal),
     achievement(?mistake,?interfering-goal),
     disables(?mistake,?desirable-state),
     goal-conflict(?goal,(GOAL actor ?y
                              obj ?w)),
     achievement(?mistake,(GOAL actor ?y
                               obj ?w)),
     not-obvious-result(?mtrans,?mistake).
```

TAU-ULTERIOR represents the situation in which y tells x information that motivates x to perform an act which results in the disablement of one of x's goals, while at the same time achieving a goal of y's which x did not forsee.

One constraint above needs some additional explanation. The last constraint, **not-obvious-result(?mtrans,?mistake)**, states that there is no short causal inference chain from the information told (?mtrans) to the disabling act (?mistake). In the current implementation of this model a "short" inference chain is one formed by the application of a single inference rule.

If we examine TAU-CONF-ENABL, we see the way **The Fox and the Crow** instantiates both this TAU and TAU-ULTERIOR, where the following variables match the same concepts:

?standing-goal --> to keep the cheese
?interfering-goal --> to show-off
?mistake --> singing.

The other concept in the **binding-spec** of TAU-ULTERIOR is,

?mtrans --> the Fox telling the Crow she has a nice voice

Since all of the concepts from TAU-CONF-ENABL are shared in TAU-ULTERIOR, we say the TAU-ULTERIOR **contains** TAU-CONF-ENABL. This containment relation between TAU-CONF-ENABL and TAU-ULTERIOR allows us to create a new TAU, a **specialization** of TAU-ULTERIOR. The new TAU is formed by taking the extra constraints from TAU-CONF-ENABL ("extra" meaning those not already in TAU-ULTERIOR) and conjoining them with those in TAU-ULTERIOR. This creates new planning advice specific to possession goals and ulterior motives. In the process of combining constraints, those from the contained TAU which are subsumed by those in the containing TAU are removed.

4 Creating New TAUs through Combination

As we saw above, we can get a non-trivial specialization of a TAU by discovering containment in a particular situation. In general TAU combination, neither TAU contains the other. In these cases we must examine the relationships or constraints among the concepts not shared between the two TAUs. For an example, consider TAU-VANITY. The representation for this TAU is given below.

```
TAU name TAU-VANITY
    binding-spec
      [?belief (KNOW actor ?x
                     obj (APPRAISAL obj (BODY-PART owner ?x)
                                    value GOOD))]
      [?standing-goal (GOAL actor ?x
                            status FAILED)]
      [?interfering-goal (GOAL actor ?x)]
      [?mistake (ACT actor ?x)]
    constraints
      intention(?interfering-goal,?plan),
      realization(?plan,?mistake),
      enablement((APPRAISAL obj (BODY-PART actor ?x)
                            value GOOD),
                 ?mistake),
      thwarting(?mistake,?standing-goal).
```

TAU-VANITY represents a situation in which x believes he has a special skill* and is thus motivated to have a goal (of "showing off" in the Fox and Crow story) which will interfere with pre-existing goals.

TAU-ULTERIOR shares a number of concepts with TAU-VANITY. The concepts which the two TAUs share in **The Fox and the Crow** are:

?standing-goal
?mistake
?interfering-goal

A novel TAU we can learn by combining TAU-VANITY and TAU-ULTERIOR is

* The representation for x's vain self-appraisal is simplified in this representation.

TAU-SUCKERED. The representation for TAU-SUCKERED is:

```
TAU name TAU-SUCKERED
      binding-spec
        [?standing-goal (GOAL actor ?x
                              status FAILED)]
        [?interfering-goal (GOAL actor ?x)]
        [?act (ACT actor ?y)]
        [?mistake (ACT actor ?x)]
      constraints
        intention(?interfering-goal,?plan),
        realization(?plan,?mistake),
        thwarting(?mistake,?standing-goal),
        achievement(?mistake,?interfering-goal),
        enablement(?plan,?sub-goal),
        resulting(?act,?state),
        achievement(?state,?sub-goal),
        goal-conflict(?standing-goal,(GOAL actor ?y
                                            obj ?w))
        achievement(?mistake,(GOAL actor ?y
                                   obj ?w))
        not-obvious-result(?act,?mistake).
```

TAU-SUCKERED embodies the planning failure of allowing someone else to take advantage of your dormant goals by providing one of the missing enablement conditions on that goal. In the case of **The Fox and the Crow** the dormant goal is the Crow's goal to show off. The missing enablement condition is a receptive audience. The Fox provides that audience and so tricks the Crow into defeating her standing goal of keeping the cheese.

TAU-SUCKERED is a new counterplanning technique which can be used by a planning program in situations where the appropriate constraints have been met.

It is intuitively obvious to people that this TAU-SUCKERED represents the proper lesson to be learned by a reader who knows about vanity and physical enablements, but who has never seen this kind of trickery before. The key issue here is how a program can learn this planning lesson also.

Clearly there are many ways (structurally) of combining TAU-VANITY and TAU-ULTERIOR. Only one (or perhaps a few) will turn into an "interesting" TAU such as TAU-SUCKERED. If so, then how is it that just the right enablement goal gets combined? The process of combining TAU is called "justification". There are a set of heuristics for justification given in next section.

In order to dynamically construct TAU-SUCKERED, there are two major steps. The first is to find the concepts which are common to the **binding-specs** of both TAU-VANITY and TAU-ULTERIOR, namely:

?standing-goal,
?interfering-goal, and
?mistake.

These concepts are included in the **binding-spec** for the new TAU, lines 2-4 and 8 in TAU-SUCKERED. Also, constraints that involve only these concepts and free variables are included in the **constraints** part of the new TAU, lines 8-11 in TAU-SUCKERED. The second step is to take the remaining concepts from the **binding-specs** and justify them; that is, make sure they are accounted for either in the new **binding-spec** or in the **constraints**.

5 Justification

In this example the concepts which are not shared are **?desirable-state** and **?belief** from TAU-VANITY and **?mtrans** from TAU-ULTERIOR. Justification takes a concept from one TAU and finds out where it fits in the causal structure for the instantiation of the other TAU.

First **?desirable-state** is found to be already subsumed by constraint number 10 of TAU-SUCKERED. This is so because an ACT which thwarts a goal mays negate a state that previously achieved the goal.

```
ACT--results-->STATE1
                 !
             negates
                 !
              STATE2--achieves-->GOAL

           !!
           \/

    ACT--thwarts-->GOAL
```

In this case, the crow's singing opens her mouth which negates her prior control over the cheese.

The achievement condition is found in constraint 10 of TAU-ULTERIOR. The heuristic which is used here is,

IF a constraint can possiblly subsume concepts,
TRY matching the constraints on subsumed concepts
against those in the component TAUs.

All that is left is to justify **?belief** and **?mtrans**. Intuitively what we want to see is a causal relation established between **?belief**, the Crow thinking she has a good voice, and **?mtrans**, the Fox's flattery of the Crow. Further we want to see **?belief** as an enablement condition on the Crow's dormant goal to show-off.

One thing to note here is that the **intention** relation is transitive. The rule for **inten-**

tion transitivity is represented as,

```
GOAL--intends-->PLAN--enables-->sub-GOAL--intends-->sub-PLAN

                    ||
                    \/

         GOAL--intends-->sub-PLAN
```

The system knows which relations are transitive, and can look up the particular kind of transitivity for a relation, i.e. what the intermediate relations are. Thus, another heuristic for justifying concepts is:

IF a transitivity rule was applied in satisfying
   a constraint of 'TAU1'
TRY opening up the transitive path and look for
      concepts of 'TAU2' that need to be justified

In this case we see that the transitive path is in constraint (line 10) from TAU-VANITY which is reproduced as constraint (line 8) in TAU-SUCKERED, so a new constraint (line 12) is added to TAU-SUCKERED for sub-GOAL. In the story this represents the fact that, in order to show off, a character must believe he has something to show-off; the Fox flattering the Crow provides this enablement condition. The satisfaction of this enablement condition is given in lines 14-15 of TAU-SUCKERED.

The two constraints in lines 14-15 represent a causal chain, and as such the first concept is included in the **binding-spec** (line 5 of TAU-SUCKERED). The two constraints in lines 15-18 are included because they directly link two concepts from the **binding-spec**. The last constraint in TAU-SUCKERED is included from TAU-ULTERIOR because **?act** and **?mistake** are in the **binding-spec**.

The result of this constraint analysis is the creation of a new TAU, TAU-SUCKERED, which represents someone being fooled into having a goal failure by being motivated to satisfy another goal triggered by vanity. This TAU serves as a new indexing structure for the Fox and the Crow story and is now available for use in future planning and comprehension tasks.

6 Progress and Future Work

A program, CRAM, is under development as part of this research. Currently, CRAM is able to understand stories that are input as unconnected Conceptual Dependency [Schank 19xx] structures. CRAM finds the planning errors in each story and characterizes them in terms of one or more TAUs. These TAUs are then used to index the story in memory for later retrival. Based on an analysis of 17 Aesop's fables, we have encoded 12 TAUs for CRAM. In addition to **The Fox and the Crow** CRAM has been tested on 2 other stories: (1) another story on deception, **The Fox and the Bear**, and (2) a variation of **The Dog and his Shadow**.

The next step in the on-going development of the theory presented here is to give CRAM enough real-world knowledge so that it can understand all 17 stories used in our analysis; then remove several of TAUs from its knowledge-base and replace them with more primitive TAUs to have it learn the more complex ones.

In addition, CRAM will be able to give advice to correct the character's planning errors. Also planned for CRAM are a natural language parser and generator so that CRAM can take in stories as verbatim input and later generate English explanation of new TAUs it has discovered.

7 Conclusions

The approach presented here allows both specialization learning and chunking learning of planning errors in multiple planning agent domains. The structures learned can be used both for critiquing plans and also for generating counterplanning advice.

References

Bewick, T., Illustrator, *Treasury of Aesop's Fables*, Avenel Books, New York, 1973.

Carbonell, J. G., *Subjective Understanding: a Computer Model of Belief Systems, Yale Ph.D Dissertation, Technical Report 150, 1979.*

DeJong, G., "Acquiring Schema Through Understanding and Generalizing Plans", in Proceedings of the Eighth International Joint Conference on Aritifical Intelligence, 1983.

Dolan, C. P., *Memory Based Processing for Cross Contextual Reminding: Reminding and Analogy Using Thematic Structures*, UCLA Computer Science Department, M.S. Thesis, 1984.

Dyer, M. G., *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension*, The MIT Press, Cambridge, Mass., 1983.

Dyer, M. G., "Understanding Stories through Morals and Remindings", in *Proceeding of the Eighth International Joint Conference on Artificial Intelligence*, 1983.

Franklin, B., *Poor Richard: The Almanacks for the Years 1733-1758*, Heritage Press, New York, 1964.

Kolodner, J. L., *Retrieval and Organization Strategies in Conceptual Memory: A Computer Model*, Technical Report 187, Yale University, Department of Computer Science, Ph.D. Disseration, 1980.

Kowalski, R., *Logic for Problem Solving*, North Holland, 1979.

Laird, J. E., Rosenbloom, P. S., and Newell, A., "Towards Chunking as a General Learning Mechnaism", in *Proceeding of the National Conference on Artificial Intelligence*, 1984.

Lebowitz, M., *Generalization and Memory in an Integrated understanding System*, Technical Report 186, Yale University, Department of Computer Science, Ph.D. Dissertation, 1980.

Meehan, J., *The Metanovel: Writing Stories by Computer*, Technical Report 74, Yale University, Computer Science Department, Ph.D. Dissertation, 1979.

Schank, R. C., "Conceptual Dependency: A Theory of Natural Language Understanding", *Cognitive Psychology*, Vol 3, No 4, 1972.

Schank, R. C. and Abelson, R. P., *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

Schank, R. C., *Dymanic Memory: A Theory of Reminding and Learning in Computers and People*, Cambridge University Press, Cambridge, 1982.

Steele, G. L., *The Definition and Implementation of a Computer Language Based on Constraints*, MIT, Ph.D Dissertation, 1980.

Sussman G. J., *A Computational Model of Skill Aquisition*, AI Technical Report 297, AI Laboratory, MIT, Ph.D. Dissertation, 1973.

Wilensky, R. *Understanding Goal-Based Stories*, Technical Report 140, Yale University, Department of Computer Science, 1978, Ph.D. Dissertation.